并发编程的挑战

2021年10月2日 星期六 13:44

挑战 { 上下文切换. 死锁. 资源限制.

① 并发一定比串行快?
不定，规模不大时，线程创建和上下文切换的开销，会消耗更多资源.

测试工具 { 上下文切换时长. Lmbench3. 上下文切换次数. vmstat.

如何减少上下文切换?
① 无锁并发编程. 锁竞争会引起上下文切换. (为什么? 请求不到锁，当前线程会切换为等待状态.)
② CAS. 不需要加锁.
↘ 确保某一变量没有被修改过.
CAS存在的问题. ① ABA → 每次更新的时候把版本号加1. JDK1.5 AtomicStampedReference已解决
② 自旋消耗. → 超过一定时间或次数退出. 如果JVM支持Pause指令，性能会提升.
③ 只能单变量 → ① 加锁 ② 封装成对象.
③ 使用最少线程. 避免浪费不必要资源
④ 协程 单个线程内实现多任务调度和切换

② 避免死锁的常见方法.
① 一个线程避免同时获取多个锁.
② 尽量每个锁占用一个资源.
③ tryLock(). 定时锁.
④ 类对数据库锁：加锁、解锁在一个连接里.

③ 资源限制. 硬件 { 带宽. I/O速度. CPU速度. } 软件 { 数据库连接数. socket连接数. }
硬件 → 解决方案 集群并行执行.
软件 → 解决方案：用资源池将资源复用.

在资源限制情况下，如何让程序更快?
根据不同的资源限制调整并发度!!!